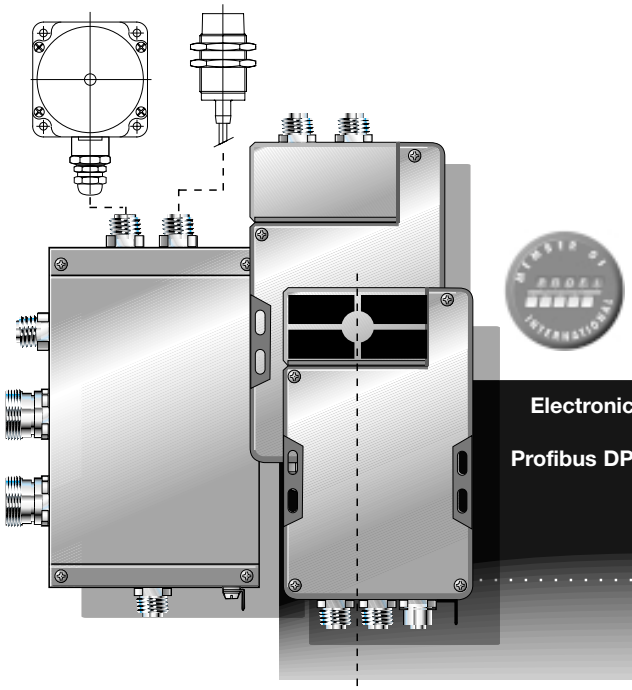
1

BALLUFF



Manual

Electronic Identification Systems BIS
Processor BIS C-60_2
Profibus DP with Memory Optimization

Deutsch – bitte wenden!

2

No. 823 024 D/E • Edition 0301
Subject to modification.
Replaces edition 0108.

http://www.balluff.de

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 (0) 71 58/1 73-0
Fax +49 (0) 71 58/50 10
E-Mail: balluff@balluff.de

Contents

Safety Considerations 4
Introduction BIS C Identification Systems 5-7
BIS C-60_2 Processor, Basic knowledge for application 8
Optimized code tag memory management 9-11
BUS interface PROFIBUS-DP 12-14
Compatibility with BIS C-6_2 processor 15
Function Description: Communication with the processor 16
Input and Output Buffers 17/18
Output buffer, configuration and explanation 19-22
Input buffer, configuration and explanation 23-26
Parametering the BIS C-60_2 processor 27-29
Processing code tags 30-33
Reading and writing 30
Reading and writing in dynamic mode 31
Reading and writing with program 31
Mixed Data Access 32/33
Monitoring code tag initialization 34
Splitting code tag memory 34
Examples for protocol sequence 35-45
Read/Write Times 46/47
LED Display 48

	BIS C-6002	BIS C-6022
Mounting Head / Processor	49	62
Opening the processor	50	63
Installing the connection cables / Mounting the PG connection	51/52	
Interface information / Wiring diagrams	53-56	64-67
Changing the EEPROM	57	68
Technical Data	58-60	69/70
Ordering information	61	71

Appendix, ASCII Table 72

Safety Considerations

Approved Operation

Series BIS C-60_2 processors along with the other BIS C system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.

Installation and Operation

Installation and operation should be carried out by trained personnel only. Unauthorized work and improper use will void the warranty and liability.

When installing the processor, follow the chapters containing the wiring diagrams closely. Special care is required when connecting the processor to external controllers, in particular with respect to selection and polarity of the signals and power supply.

Only approved power supplies may be used for powering the processor. See chapter 'Technical Data' for details.

Use and Checking

Prevailing safety regulations must be adhered to when using the identification system. In particular, steps must be taken to ensure that a failure of or defect in the identification system does not result in hazards to persons or equipment

This includes maintaining the specified ambient conditions and regular testing for functionality of the identification system including all its associated components.

Fault Conditions

Should there ever be indications that the identification system is not working properly, it should be taken out of commission and secured from unauthorized use

Scope

This manual applies to processors in the series BIS C-6002-028-...-03-KL2 and BIS C-6022-028-050-03-ST....

Introduction
BIS C Identification Systems

This manual is designed to assist the user in setting up the control program and installing and starting up the components of the BIS C Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS C Identification Systems belongs in the category of

non-contact systems for reading and writing.

This dual function permits applications for not only transporting information in fixed-programmed code tags, but also for gathering and passing along up-to-date information as well.

If 2 read/write heads are connected to a BIS C-60_2 processor, both heads can be operated independently of each other. This means for example that you can read a code tag from one head while writing to another code tag at the other head.

Applications

Some of the notable areas of application include

- **for controlling material flow in production processes** (e.g. in model-specific processes)**,** for workpiece conveying in transfer lines, in data gathering for quality assurance , for gathering safety-related data,
- **in tool coding and monitoring;**
- **in equipment organization;**
- **in storage systems for monitoring inventory movement;**
- **in transporting and conveying systems;**
- **in waste management for quantity-based fee assessment.**

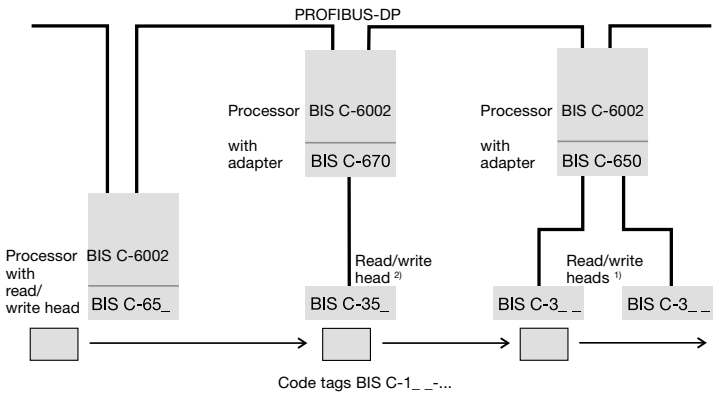
Introduction
BIS C Identification Systems

System Components

The main components of the BIS C Identification Systems are:

- **Processor,**
- **Read/Write Heads and**
- **Code Tags**

Configuration with BIS C-6002 processor

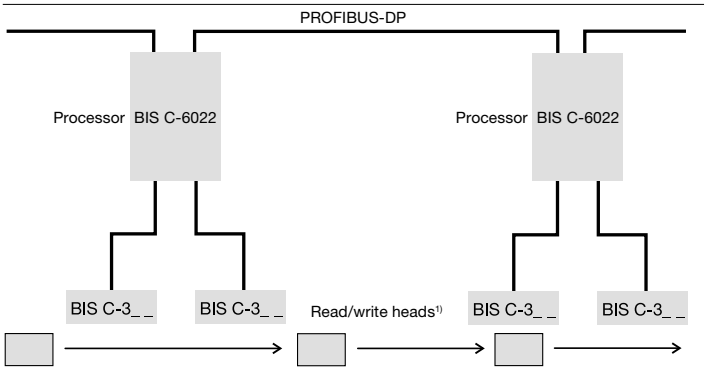


Schematic representation of an Identification System (example)

1) BIS C-3_ _ series, except BIS C-350 and -352 2) only BIS C-350 or -352

Introduction
BIS C Identification Systems

Configuration with BIS C-6022 processor



Schematic representation of an identification system (example)

1) BIS C-3_ _ series, except BIS C-350 and -352

BIS C-60_2 Processor
Basic knowledge for application

Selecting System Components

The **BIS C-6002** processor has a plastic housing. Connections are made through a terminal strip, with the cables secured by PG fittings. A single read/write head from BIS C-65_ series can be directly mounted to the processor, which creates a compact unit. If the BIS C-650 adapter is attached instead of the BIS C-65_ read/write head, two read/write heads may be cable connected. If the BIS C-670 adapter is attached, one read/write head may be cable connected.

The **BIS C-6022** processor has a metal housing. Connection is made through round connectors. Two read/write heads can be cable connected to the BIS C-6022 processor.

Series BIS C-60_2 processors have in addition a digital input. The input has various functions depending on the configuration (see Parametering).

Whether the compact version of the processor with integrated read/write head makes sense or whether the external solution is preferred depends primarily on the spatial arrangement of the components. There are no functional limitations. All read/write heads are suitable for both static and dynamic reading. Distance and relative velocity are based on which code tag is selected. Additional information on the read/write heads in series BIS C-65_ and series BIS C-3_ _ including all the possible code tag/read-write head combinations can be found in the manuals for the respective read/write heads.

The system components are electrically supplied by the processor. The code tag represents an free-standing unit and needs no line-carried power. It receives its energy from the read/write head. The latter constantly sends out a carrier signal which supplies the code head as soon as the required distance between the two is reached. The read/write operation takes place during this phase. Reading and writing may be dynamic or static.

9

BIS C-60_2 Processor
Optimized code tag memory management

Protocol

The BIS C-60_2-028 version works with a further developed Balluff protocol, which optimizes memory handling in the code tag. This version would be required whenever the required number of write cycles exceeds the number permitted for the EEPROM in the code tag. Depending on type, this will lie between 100,000 and 1,000,000 cycles and is indicated on the data sheet for each respective code tag. You must select a code tag whose memory size is a multiple of the number of bytes actually used for writing.

For any given code tag the maximum number of write cycles permitted can be calculated as follows:

$$W = W_{CT} * \frac{K_{CT} - 5}{n + 4}$$

K_{CT} = Memory size of code tag in bytes
W_{CT} = maximum number of write cycles for code tag per data sheet
n = maximum number of bytes for read/write commands

Using this intelligent method of memory handling, the number of write cycles is stored in a counter. When the permitted number is reached, the previously used memory range is blocked and a new memory range is used. This process is continued until the last memory range is used up. During the last 1,000 write cycles the processor acknowledges all successful write operations with a 'pre-warning' (56_{Hex} in subaddress 02_{Hex} of the input buffer).

When the maximum number of write cycles for the last memory range is reached, writing to this memory range continues but an additional 'final warning' is indicated (45_{Hex} in subaddress 02_{Hex} of the input buffer).

10

BIS C-60_2 Processor
Optimized code tag memory management

Initializing the code tag

In order to use a code tag for memory optimization, the code tag must first be initialized. This can be done either with the Handy Terminal BIS C-800, or from a PC station with a built-in BIS C-480-007-PC card, or using the initializing command of BIS C-60_2-028.

The first 5 bytes of the memory are used for the designator:

Byte No.	Initialization (hexadecimal)	Meaning / Function
0	00H	Number of currently used memory range on code tag
1	01H	128 Byte Enter memory range of code tag
	02H	256 Byte
	03H	512 Byte
	04H	1 024 Byte
	05H	2 048 Byte
	06H	4 096 Byte
	07H	8 192 Byte
	08H	16 384 Byte
	09H	127 Byte
	0AH	255 Byte
	0BH	511 Byte
	0CH	1 023 Byte
	0DH	2 047 Byte
2	01H	100,000 Enter maximum number of write cycles of the code tag
	02H	200,000 (see data sheet)
	03H	300,000
	04H	400,000
	05H	500,000
	0AH	1,000,000
3 and 4	maximum 0400H	maximum number of bytes per read/write cycle = 1 kByte (segment size)

11

BIS C-60_2 Processor
Optimized code tag memory management

Initializing the code tag (continued)

Example of an initialization:

On a code tag with 128 byte memory size, 24 bytes shall be used for reading and writing. The maximum permissible number of write cycles according to the code tag data sheet is 100,000. The result is the following initialization of the first 9 bytes of memory:

00_H 01_H 01_H 00_H 18_H 00_H 00_H 00_H 00_H

- Number of the 1st memory range
- Memory size of code tag 128 bytes
- Permissible no. of write cycles 100,000
- Maximum no. of bytes per read/write cycle set to 24 bytes
- Counter of 1st field set to 0

An address named in a telegram does not distinguish between code tags with and without initialization.

The ratio of memory size to memory requirement thus permits 400,000 write cycles, since the available memory size can be divided into 4 memory ranges of 100,000 bytes each. The pre-warning is given after the 399,000th write cycle.

Control Function

The processor writes data from the host system to the code tag or reads data from the tag through the read/write head and prepares it for the host system. Host systems may include:

– **a host computer (e.g. industrial PC) or**
– **a programmable logic controller (PLC)**

12

BUS interface PROFIBUS-DP

PROFIBUS-DP

Communication between the BIS C-60_2 processor and the host system is via PROFIBUS-DP.

The PROFIBUS-DP system consists of the components:

– the bus master and
– the bus modules/slaves (here the BIS C-60_2 processor).

Important hints for use with PLC:

In some control systems the PROFIBUS-DP data area is not synchronously transmitted with the updating of the input/output content. If more than 2 bytes of data are sent, a mechanism must be used which guarantees that the data in the PLC and the data in the BIS C are always identical!

1st alternative: Synchronous data transmission as a setting on the Master
In this method the bus Master ensures that all the data necessary for the respective Slave are always sent contiguously. There is usually a special software function in the PLC which likewise controls access between the PLC and bus Master so that data are always sent contiguously.

2nd alternative: Set 2nd bit header
Data exchange between PLC and BIS is controlled by the so-called bit header. This is always the first byte of the respective read/write head in the data buffer. This bit header exists both in the input range (data from BIS to the PLC) and in the output range (data from the PLC to the BIS). Ilf this bit header is also sent as the last byte, a comparison of these two bytes can be used to guarantee the consistency of the transmitted data.

In this method the PLC cycle is unaffected nor is the bus access time changed. All that is required is that a byte in the data buffer be used for the 2nd bit header instead of for user data.

This 2nd alternative is the Balluff recommended setting (factory default).

BUS interface PROFIBUS-DP

Unit's Master Data

For the correct parametering of the bus master as per type, a diskette, containing the unit's master data in the form of a GSD file is included with the BIS C-60_2 processor.

Station Address

The Processor BIS C-60_2 is delivered with the station address 126. This has to be set individually before using in a bus system. See information on 🗋 14.

Input/Output Buffer

An input buffer and an output buffer are used for the data exchange with the control system. The size of these buffers has to be configured via the master.

The possible settings are entered in the GSD file (and Type file). A minimum of 4 and a maximum of 128 bytes can be accommodated. However, it must be an even number.

Parametering Bytes
User-Parameter Bytes

Besides, in the case of the BIS C-60_2 processor, there are 6 further bytes (User-Parameter Bytes) which have to be set while parametering. The significance of the 6 bytes for parametering is described starting from 🗋 28.

The preset is stored in the GSD file.

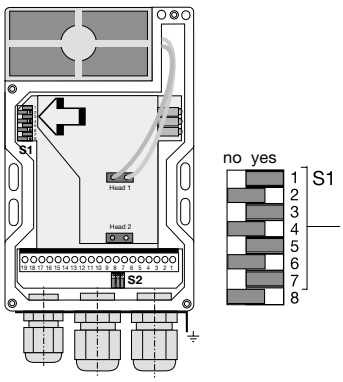
BUS interface PROFIBUS-DP

Station Address setting

The station address under which the unit is accessed on the bus can be assigned through the slide switch S1. Each address shall be assigned only once.

The slide switch S1 is binary coded. The setting of the station address is carried out according to the scheme shown in the table. Switch position: no = left, yes = right.

The address 85 is set in the following figure.



Slide switch S1 (with cover removed)

Station Address	Slide switch S1						
	7	6	5	4	3	2	1
	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	not allowed						
1	yes	no	no	no	no	no	no
2	no	yes	no	no	no	no	no
3	yes	yes	no	no	no	no	no
4	no	no	yes	no	no	no	no
5	yes	no	yes	no	no	no	no
...							
85	yes	no	yes	no	yes	no	yes
...							
123	yes	yes	no	yes	yes	yes	yes
124	no	no	yes	yes	yes	yes	yes
125	yes	no	yes	yes	yes	yes	yes
126	no	yes	yes	yes	yes	yes	yes
127	not allowed						

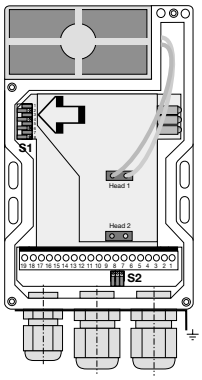
To open the cover of the processor, see 🗋 50 for BIS C-6002 or 🗋 63 for BIS C-6022.

Compatibility with BIS C-6_2 processor

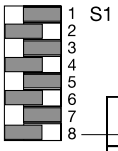
Setting compatibility

Slide switch S1 is used to set compatibility with the BIS C-602 and BIS C-622 processors.

If the BIS C-60_2 processor is set to be compatible with the BIS C-602 or BIS C-622, all settings for data exchange must be made as described in the sections on parametering, function description, protocol sequence and LED display in the user's manual for the BIS C-6_2 processor! This user's manual can be mailed on request, or you may download it from the Internet at `www.balluff.de`.



Slide switch S1 (with cover removed)



In "Compatible with BIS C-6_2" mode, the error code also has the error number 19_{Hex} added. This error number indicates that a command used in this mode does not work.

Slide switch S1	
8	compatible with
yes	BIS C-6_2
no	BIS C-60_2

Key: no = switch left
yes = switch right

In the illustration compatibility with the BIS C-6_2 is not set.

To open the cover of the BIS C-6002 processor, see 50, and for BIS C-6022 see 63.

Function Description
Communication with the processor

Basic Procedure

Communication between the host system and the processor takes place using a fixed protocol sequence. Data integrity from the control to the processor and vice-versa is indicated by a control bit. This bit is used to implement a handshake between the control and the processor.

Following is a simplified representation of the sequence of a job sent from the control to the processor:

1. The control sends a command designator to the processor together with the associated command parameters and sets a bit (AV bit). This bit indicates to the processor that the transmitted data are valid and that the job is now beginning.
2. The processor takes the job and sets a bit (AA bit), which indicates this to the control.
3. If an additional exchange of data between the control and the processor is required to carry out the job, each uses a bit (TI bit and TO bit) to indicate that the control / processor is now ready for additional data exchange or has accepted the received data.
4. Once the processor has carried out the job correctly, it sets a bit (AE bit).
5. Once the control has accepted all the important data, it indicates this to the processor by resetting the bit that was set at the beginning (AV bit).
6. The processor now in turn sets all the control bits that were set during the sequence (AA bit, AE bit) and is ready for the next job.

Please see also 30...34 and the examples on 35...45.

Function Description
Input and Output Buffers

Input and Output Buffers

In order to transmit commands and data between the BIS C-60_2 and the host system, the latter must prepare two fields. These two fields are:

– **the output buffer**
for the control commands which are sent **to** the BIS Identification System and
for the data to be written.

– **the input buffer**
for the data to be read and
for the designators and error codes which come **from** the BIS Identification System.

The possible setting values are stored in the GSD file.

The buffer size can be selected between 4 and 128 bytes in steps of 2 bytes. This must be given by the master during parametering. The total buffer size is divided into 2 ranges:

Buffer range 1 for Read/Write Head 1; size is specified in paramter byte 6.
Buffer range 2 for Read/Write Head 2; size = total buffer size – buffer size of Read/Write Head 1. See 18 for example.

If a buffer size of less than 6 bytes (8 bytes with double bit header) is used for a read/write head, no read/write job can be carried out.

Buffer size – 1 = number of bytes read without double bit header;
Buffer size – 2 = number of bytes read with double bit header.

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Input and Output Buffers

Input and Output Buffers
(continued)

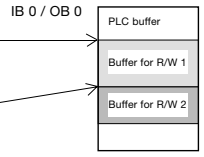
Example: The 82 bytes for the total buffer need to be distributed. An input/output buffer of 46 bytes is assigned to Read/Write Head 1. This results in an input/output buffer of 36 bytes for Read/Write Head 2.

Procedure: The buffer size for Read/Write Head 1 is set to 46 bytes. This means using the parameter byte 6 to enter Hex value 2E (corresponds to 46 decimal), which corresponds to binary 0**01011**10.

PLC Organisation: The buffer range starts at input byte IB 32 and output byte OB 32.

Result:

Read/Write Head 1: (R/W 1)	Subaddress 00	IB 32 and OB 32
	Input buffer	IB 32 to IB 77
	Output buffer	OB 32 to OB 77
Read/Write Head 2: (R/W 2)	Subaddress 00	IB 78 and OB 78
	Input buffer	IB 78 to IB 113
	Output buffer	OB 78 to OB 113



Note that these buffers can be in two different sequences depending on the type of control.

The following description is based on sequence 1!

Sequence 1	Sequence 2
Subaddress 00	Subaddress 01
01	00
02	03
03	02
04	05
05	04
06	07
07	06

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Output buffer, configuration and explanation

Configuration of the output buffer for one (1) read/write head

The last byte can be arranged as a 2nd bit header through parametering (default).

Subaddress \ Bit No.	7	6	5	4	3	2	1	0	
00Hex = Bit Header	CT	TI				GR		AV	Bit Name
01Hex	Command Designator					or	Data		
02Hex	Start Address (Low Byte) or Program No.					or	Data		
03Hex	Start Address (High Byte)					or	Data		
04Hex	No. of Bytes (Low Byte)					or	Data		
05Hex	No. of Bytes (High Byte)					or	Data		
06Hex	Data								
...	Data								
Last Byte	2nd Bit Header (as above)					or	Data		

Description of Output Buffer

Sub-address	Bit Name	Meaning	Function Description
00Hex Bit Header	CT	Code tag type 0 1	Select code tag type: for code tag type: 32 Byte block size BIS C-1_ _-02, -03, -04, -05 64 Byte block size BIS C-1_ _-10, -11, -30
	TI	Toggle-Bit In	Shows during a read action that the controller is ready for additional data.
	GR	Ground state	Causes the BIS system to go to the ground state for the respective read/write head. Any pending command is cancelled.
	AV	Command	Signals the identification system that a command for the respective read/write head is present.

(continued next)

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
01Hex	Command designator	
	00Hex	No command present
	01Hex	Read code tag
	02Hex	Write to code tag
	06HEX	Store program in the EEPROM for the Mixed Data Access function
	08HEX	Initialize code tag
	09HEX	Save monitoring of code tag initialization in EEPROM
	0AHEX	Save split address for dividing the code tag into an area having memory optimization in EEPROM
	21HEX	Read for Mixed Data Access function (corresponding to the program stored in the EEPROM)
	22HEX	Write for Mixed Data Access function (corresponding to the program stored in the EEPROM)
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.

(continued next)

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
02Hex	Start address (Low Byte)	Address at which reading from or writing to the code tag begins (the Low Byte includes the address range from 0 to 255). (Das Low Byte deckt den Adressbereich von 0 bis 255 ab).
or:	Program No	Number of the program to be stored in the EEPROM in conjunction with command ID 06Hex for Mixed Data Access function (values between 01Hex and 0AHex are allowed!).
or:	Program No.	Number of the program stored in the EEPROM for read or write operations in conjunction with command ID 22Hex or 22Hex for the Mixed Data Access function.
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
or:	Parametering data	for writing to the EEPROM.
03Hex	Start address (High Byte)	Start address for reading from or writing to the code tag (the High Byte is additionally used for the address range from 256 to 8,191)
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
or:	Parametering data	for writing to the EEPROM.

(continued next 📄)

Please note the basic procedure on 📄 16 and 30...34 and the examples on pages 📄 35...45.

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
04Hex	No. of bytes (Low Byte)	Number of bytes to read or write beginning with the start address (the Low Byte includes from 1 to 256 bytes).
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
05Hex	No. of bytes (High Byte)	Number of bytes to read or write beginning with the start address (the High Byte is additionally used for the range between 257 and 8,192 bytes).
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
06Hex	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
...	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
Last byte		
	2nd Bit header	The data are valid if the 1st and 2nd bit header are identical.
or:	Data	for writing to the code tag.
or:	Program data	for writing to the EEPROM.

Please note the basic procedure on 📄 16 and 30...34 and the examples on pages 📄 35...45.

Function Description
Input buffer, configuration and explanation

Configuration of the input buffer for one (1) Read/Write head

The last byte can be arranged as a 2nd bit header through parametering (default).

Subaddress \ Bit No.	7	6	5	4	3	2	1	0	
00_{Hex} = Bit Header	BB	HF	TO	IN	AF	AE	AA	CP	Bit Name
01_{Hex}	Error Code					or	Data		
02_{Hex}	Data								
03_{Hex}	Data								
04_{Hex}	Data								
05_{Hex}	Data								
06_{Hex}	Data								
...	Data								
Last byte	2nd Bit Header (as above)					or	Data		

Description of Input Buffer

Sub-address	Bit Name	Meaning	Function Description
00_{Hex}	BB	Ready	The BIS Identification System is in the Ready state.
Bit Header	HF	Head Error	Cable break from read/write head or no read/write head connected.
	TO	Toggle-Bit Out	for read: BIS has new/additional data ready. for write: BIS is ready to accept new/additional data.

(continued on next page)

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Bit Name	Meaning	Function Description
00_{Hex}	(continued)		
Bit Header	IN	Input	If the parameter "Input IN" is 1, this bit indicates the state of the Input.
	AF	Command Error	The command was incorrectly processed or aborted.
	AE	Command end	The command was finished without error.
	AA	Command start	The command was recognized and started.
	CP	Codetag Present	Code tag present within the active zone of the read/write head.
			In addition to the CP bit, the output signal **CT-Present** is available. This allows you to process the presence of a code tag directly as a hardware signal.

Sub-address	Meaning	Function Description
01_{Hex}	Error code	Error number is entered if command was incorrectly processed or aborted. Only valid with AF bit!
	00_{Hex}	No error.
	01_{Hex}	Reading or writing not possible because no code tag is present in the active zone of a read/write head.
	02_{Hex}	Read error.
	03_{Hex}	Code tag was removed from the active zone of the read/write head while it was being read.
	04_{Hex}	Write error.

(continued on next page)

Please note the basic procedure on 16 and 30...34 and the examples on pages 35...45.

Function Description
Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Meaning	Function Description
01Hex	Error code (continued)	
	05Hex	Code tag was removed from the active zone of the read/write head while it was being written.
	06Hex	Access error in memory.
	07Hex	AV bit is set but the command designator is missing or invalid.
	or:	Number of bytes is 00Hex.
	09Hex	Cable break to select read/write head, or head not connected.
	0CHex	The EEPROM cannot be read/programmed.
	0DHex	Faulty communication with the code tag.
	0FHex	Contents of the 1st and 2nd bit header (1st and last bytes) of the output buffers are not identical (2nd bit header must be served).
	13Hex	Start address + number of bytes > memory range entered in the initialization.
	14Hex	invalid max. number of write cycles entered in the initialization.
	15Hex	invalid memory size of the code tag entered in the initialization.
	16Hex	Max. 1 kB segment size exceeded.
	17Hex	Code tag incorrectly initialized (comparison with nominal data with command identifier 09Hex does not bring agreement).
	18Hex	Code tag not initialized. All bytes are still 00Hex.
or:	Data	Data which was read from the code tag.

(continued on next page)

Please note the basic procedure on pages 16 and 30...34 and the examples on pages 35...45.

Function Description
Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Meaning	Function Description
02Hex	Data	Data which was read from the code tag,
or:	Warning	Memory management (details see page 9) signals:
	56Hex	Pre-warning (1,000 write cycles left).
	45Hex	End warning.
...	Data	Data which was read from the code tag.
Last byte		
	2nd Bit header	The data are valid if the 1st and 2nd bit headers are in agreement.
or:	Data	Data which was read from the code tag.

Please note the basic procedure on pages 16 and 30...34 and the examples on pages 35...45.

Function Description
Parametering the BIS C-60_2 processor

Parameters, Overview

There are 6 user parameter bytes stored on the Profibus master that can be used to activate and deactivate various functions. Setting is done directly by linking a device to the Profibus master. The parameter default settings are stored in the GSD file.

- **Dynamic operation on Read/Write Head 1 or 2:**
  If dynamic operation is parametered, a read/write job can be sent even though there is no code tag in the active zone of the head. As soon as a code tag passes by the head, the command is immediately carried out.
- **2nd bit header at end of in- and output buffer:**
  The 2nd bit header (factory setting) prevents data from being accepted by the bus as long as it is not fully updated.
- **Display state of the digital input in the bit header of the input buffer:**
  If this function is activated, the IN-bit displays the state of the digital input of the processor: IN = 0 → digital input low; IN = 1 → digital input high
- **Reset BIS C-60_2 processor through the digital input:**
  If this function is activated, the processor is reset when the digital input is set to high.
- **Monitor code tag initialization:**
  If this function is activated, the initialization data on the code tag are compared with the initialization data stored in the processor. If these do not agree, a read/write error is rejected with an error message.
- **Split code tag memory:**
  If this function is activated, data on the code tag beyond a certain specified address are no longer optimized. The split address must have first been stored in the processor using command identifier $0A_{Hex}$.

Function Description
Parametering the BIS C-60_2 processor

Parametering Bytes
User-Parameter Bytes

> For parametering all 6 bytes must always be transferred in HEX. Only the bits mentioned may be changed. No guaranty will be given for the proper functioning of the BIS C-60_2 if any of the other bits are changed.

The default values (factory setting) for the 6 bytes are:

	1st byte	**2nd byte**	3rd byte	**4th byte**	**5th byte**	**6th byte**
HEX	00	80	00	82	00	02
Binary	00000**00**	10**0**00000	00000000	**10**0000**1**0	000**00**000	**00000010**
	bit 1	**bit 5**		**bit 7** **bit 2**	**bit 4**	**bit 1...8**
	bit 2			**bit 8**	**bit 5**	

The bits which serve for parametering have the following functions:

1st byte, bit 2, Monitor code tag initialization

1st byte, bit 1 Divide code tag memory into one area with memory expansion and one area without memory expansion.

Bit state: 0 = no
1 = yes

2nd byte, bit 5, Dynamic mode on read/write head 1
(for effects on read/write times, see 📄 46/47)

(Continued on next 📄)

Function Description
Parametering the BIS C-60_2 processor

Parametering Bytes (continued)

4th byte, bit 8, Arrange a 2nd bit header at the end of the input and output buffers.

If this function is selected, then the minimum size of both buffers is 4 words (8 bytes) each.

4th byte, bit 7, Display state of the digital input in the bit header of the input buffers:
0 = no
1 = yes
Input is Low: "IN" in the bit header of the input buffers = 0.
Input is High: "IN" in the bit header of the input buffers = 1.

4th byte, bit 2, Reset the BIS C-60_2 processor through the digital input:
0 = no
1 = yes
Input is Low: Do not reset.
Input is High: Reset.

Bit state: 0 = no
1 = yes

5th byte, bit 5 Dynamic mode on read/write head 2
(for effects on read/write times, see 46/47)

6th byte, bit 1...6 No. of bytes in input and output buffer which shall be used for read/write head 1, see example on 18.

The specification for the input and output buffer on the Master applies to both read/write heads, i.e. this buffer must be divided for both heads. The specification is done in Hex format and must be in a range between 02_{Hex} and 80_{Hex} (128 dec.).

If only one read/write head (Head 1) will be used, you may enter the same value here as for the total buffer size. An entry of less than 2 bytes results in an undefined state.

Function Description
Processing code tags

Reading and writing

To carry out a read or write job, the code tag must be located in the active zone of the read/write head.

A read/write job has the following sequence (see examples on 37ff):

1. The host sends to the output buffer:
   – the command designator to subaddress 01_{Hex},
   – the start address for reading or writing to subaddress 02_{Hex}/03_{Hex},
   – the number of bytes for reading or writing to subaddress 04_{Hex}/05_{Hex},
   – the CT bit according to the code tag type (block size),
   – and sets the AV bit in the bit header to high.
2. The processor:
   – takes the request (AA in the bit header of the input buffer to high),
   – begins to transport the data;
   read = from code tag to input buffer,
   write = from output buffer to code tag.
   (Larger data quantities are sent in blocks
   block size "with 2nd bit header" = buffer size – 2),
   block size "without 2nd bit header" = buffer size – 1).
   The toggle bits in the two bit headers are used as a kind of handshaking between the host and the BIS C-60_2 processor.
3. The processor has processed the command correctly (AE bit in the bit header of the input buffer). If an error occurred during execution of the command, an error number will be written to subaddress 01_{Hex} of the input buffer and the AF bit in the bit header of the input buffer will be set.

Function Description
Processing code tags

Special characteristics

To adjust the read/write functions to the numerous possible applications, a few unique features have been implemented that the user can select and set when parametering or programming the processor. These are as follows:

Reading and writing in dynamic mode

In normal operation a read/write job is rejected by the BIS C-60_2 processor by setting the AF bit and an error number if there is no code tag in the active zone of the read/write head. If dynamic mode is configured, the processor accepts the read/write job and stores it. When a code tag is recognized, the stored job is carried out.

Read from code tag, with program Mixed Data Access

The command identifier 21_{Hex} can be used to read out the program records stored in the program from the code tag. The user must document exactly which data are to be read from where and with what number of bytes for the respective program (see example 8 on 🗋 43).

Write to code tag, with program Mixed Data Access

The command identifier 22_{Hex} can be used to write the program records stored in the program to the code tag. The user must document exactly which data are to be written from where and with what number of bytes for the respective program (see example 9 on 🗋 44).

Function Description
Processing code tags

Mixed Data Access

Small read/write programs can be stored in the BIS C-60_2 processor's EEPROM.

The Mixed Data Access function is useful when the required information is stored on the code tag at various addresses. This function makes it possible to read out this "mixed", i.e. non-contiguously stored data from the code tag in a single procedure and using just one command.

Up to 10 programs with up to 25 instructions can be stored. Each program instruction contains a "start address" and a "number of bytes" specification. The amount of data for reading may not exceed 2 kB.

Storing a program:
The command identifier 06_{Hex} is used to send the read/write program to the BIS C-60_2 processor. One program per command can be stored. All 25 program records plus an additional 2 bytes with $FF_{Hex}FF_{Hex}$ as a terminator must always be sent. This means a total of **104 bytes** of information per program must be sent (including the command identifier and program number).

The individual program records must all be contiguous. They must be sent one after the other and be terminated with $FF_{Hex}FF_{Hex}$ as a terminator. It is recommended that the remaining, unused memory sector be filled with $FF_{Hex}FF_{Hex}$.

If an address range is selected twice, the data will also be output twice.

Function Description
Processing code tags

Mixed Data Access (cont.)

The following shows the structure of a program:

Program structure	Subaddress	Value	Range
Command designator	01_{Hex}	06_{Hex}	
1. Program record			
Program number	02_{Hex}	01_{Hex}	01_{Hex} to $0A_{Hex}$
1st data record:			
Start address Low Byte	03_{Hex}		
Start address High Byte	04_{Hex}		
Number of bytes Low Byte	05_{Hex}		
Number of bytes High Byte	06_{Hex}		
2nd data record:			
...			
25th data record:			
Start address Low Byte	03_{Hex}		
Start address High Byte	04_{Hex}		
Number of bytes Low Byte	05_{Hex}		
Number of bytes High Byte	06_{Hex}		
Terminator	FF_{Hex} FF_{Hex}		

To store a second program, repeat this process.

The procedure for writing these settings to the EEPROM is described in the 7th example on 📄 40...42.

Replacing the EEPROM is described on 📄 57 for BIS C-6002 and on 📄 68 for BIS C-6022.

Function Description
Monitoring initialization, splitting memory

Monitoring code tag initialization
(see 2nd example on 📄 36)

Command identifier 09_{Hex} allows initialization data to be stored in the processor.

If the Monitor Code Tag Initialization function is activated, a read/write command results first in the initialization data on the code tag being compared with the data in the processor. If these do not agree, error message 17_{Hex} is output. If the code tag has not even been initialized yet, i.e. all initialization data are 0, error message 18_{Hex} is output.

Splitting code tag memory
(see 3rd example on 📄 36)

Command identifier $0A_{Hex}$ allows an address to be stored in the processor starting at which the code tag data are no longer memory-optimized.

When dividing the memory, make the sector with memory optimization large enough so that it is sufficient for the maximum intended number of write cycles (see 📄 9 for calculation).

Division brings an advantage in processing small quantities of data that are frequently programmed (written) in the sector not having memory optimization.

If the split function is activated, this memory division applies to every code tag entering the read/write zone of the processor.

Function Description
Examples for protocol sequence

Example No. 1

For configuring with double bit header and 8-byte buffer size!

Initializing the code tag for memory optimization
(code tag type with 32 byte block size)

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 08_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 0 (32 Byte block size), set AV-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit, invert TO-Bit

3.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the input buffer:

$01...05_{Hex}$	Copy the 5 initialization bytes
$00_{Hex}/07_{Hex}$	Set AE-Bit

5.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

6.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 2

For configuring with double bit header and 8-byte buffer size!

Store code tag initialization data for monitoring in the processor

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command identifier 09_{Hex}
$02...05_{Hex}$	Enter 2nd through 5th initialization bytes
$00_{Hex}/07_{Hex}$	Set AV-bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-bit and TO-bit
$02...05_{Hex}$	Copy 2nd through 5th initialization bytes
$00_{Hex}/07_{Hex}$	Set AE-bit

3.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Example No. 3

For configuring with double bit header and 8-byte buffer size!

Store split address in processor

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command identifier $0A_{Hex}$
$02...03_{Hex}$	Enter split address (Low/High byte)
$00_{Hex}/07_{Hex}$	Set AV-bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-bit and TO-bit
$02...03_{Hex}$	Copy split address (Low/High byte)
$00_{Hex}/07_{Hex}$	Set AE-bit

3.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 4

For configuring with double bit header and 8-byte buffer size!

Read 17 bytes starting at code tag address 10 (code tag type with 32 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte 11_{Hex}
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 0 (32 Byte block size), set AV-Bit

3.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Invert TI-Bit

5.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy second 6 data bytes

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Invert TI-Bit

7.) Process subaddresses of the input buffer:

$01...05_{Hex}$	Copy the remaining 5 data bytes

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Set AE-Bit

4.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

6.) Process subaddresses of the input buffer:

$01...05_{Hex}$	Enter the remaining 5 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

8.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 5

For configuring with double bit header and 8-byte buffer size!

Read 30 bytes starting at code tag address 10 with read error
(code tag type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte $1E_{Hex}$
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	Set CT-Bit to 1 (64 Byte block size), set AV-Bit

3.) Process subaddress of the input buffer:

01_{Hex}	Copy error number

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

If an error occurs right away:

$00_{Hex}/07_{Hex}$	Set AA-Bit
01_{Hex}	Enter error number
$00_{Hex}/07_{Hex}$	Set AF-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AF-Bit

Function Description
Examples for protocol sequence

Example No. 6

For configuring with double bit header and 8-byte buffer size!

Write 16 bytes starting at code tag address 20 (code tag type with 32 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 02_{Hex}
02_{Hex}/03_{Hex}	Start address 14_{Hex} / 00_{Hex}
04_{Hex}/05_{Hex}	No. of bytes 10_{Hex} / 00_{Hex}
00_{Hex}/07_{Hex}	CT-Bit to 0 (32 Byte block size), set AV-Bit

3.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the first 6 data bytes
00_{Hex}/07_{Hex}	Invert TI-Bit

5.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
00_{Hex}/07_{Hex}	Invert TI-Bit

7.) Process subaddresses of the output buffer:

$01...04_{Hex}$	Enter the remaining 4 data bytes
00_{Hex}/07_{Hex}	Invert TI-Bit

9.) Process subaddresses of the output buffer:

00_{Hex}/07_{Hex}	Reset AV-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

00_{Hex}/07_{Hex}	Set AA-Bit, invert TO-Bit

4.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Copy the first 6 data bytes

Process subaddress of the input buffer:

00_{Hex}/07_{Hex}	Invert TO-Bit

6.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Copy the second 6 data bytes

Process subaddress of the input buffer:

00_{Hex}/07_{Hex}	Invert TO-Bit

8.) Process subaddresses of the output buffer:

$01...04_{Hex}$	Copy the remaining 4 data bytes

Process subaddress of the input buffer:

00_{Hex}/07_{Hex}	Set AE-Bit

10.) Process subaddresses of the input buffer:

00_{Hex}/07_{Hex}	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 7 Store Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Storing a program for reading out 3 data records:

1st data record	Start address	5	Number of bytes	7
2nd data record	Start address	75	Number of bytes	3
3rd data record	Start address	312	Number of bytes	17

Total number of bytes exchanged in the operation: 27 bytes

All 104 bytes are written for the programming.

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 06_{Hex}
02_{Hex}	Program number 01_{Hex}
00_{Hex}/07_{Hex}	CT-Bit to 0 or 1 (depending on block size), set AV-Bit

3.) Process subaddresses of the output buffer:

01_{Hex}	1st start address	(Low Byte) 05_{Hex}
02_{Hex}		(High Byte) 00_{Hex}
03_{Hex}	1st number of bytes	(Low Byte) 07_{Hex}
04_{Hex}		(High Byte) 00_{Hex}
05_{Hex}	2nd start address	(Low Byte) $4B_{Hex}$
06_{Hex}		(High Byte) 00_{Hex}
00_{Hex}/07_{Hex}	Invert TI-Bit	

Host:

2.) Process subaddresses of the input buffer:

00_{Hex}/07_{Hex}	Set AA-Bit, invert TO-Bit

4.) Process subaddresses of the input buffer:

00_{Hex}/07_{Hex}	Invert TO-Bit

Continued on next page.

Function Description
Examples for protocol sequence

Example No. 7
Store Mixed Data Access program
(continued)

For configuring with double bit header and 8-byte buffer size!

Host:

5.) Process subaddresses of the output buffer:

01_{Hex} 02_{Hex}	2nd number of bytes	(Low Byte) 03_{Hex} (High Byte) 00_{Hex}
03_{Hex} 04_{Hex}	3rd start address	(Low Byte) 38_{Hex} (High Byte) 01_{Hex}
05_{Hex} 06_{Hex}	3rd number of bytes	(Low Byte) 11_{Hex} (High Byte) 00_{Hex}
$00_{Hex}/07_{Hex}$	Invert TI-Bit	

Host:

6.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

7.) Process subaddresses of the output buffer:

$01_{Hex}/02_{Hex}$	Terminator	FF_{Hex}/FF_{Hex}
$03_{Hex}/04_{Hex}$	(not used)	FF_{Hex}/FF_{Hex}
$05_{Hex}/06_{Hex}$	(not used)	FF_{Hex}/FF_{Hex}
$00_{Hex}/07_{Hex}$	Invert TI-Bit	

8.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

Fill all unused start addresses and number of bytes with FF_{Hex}!

Continued on next 🗋.

Function Description
Examples for protocol sequence

Example No. 7
Store Mixed Data Access program
(continued)

For configuring with double bit header and 8-byte buffer size!

Host:

35.) Process subaddresses of the output buffer:

$01_{Hex}/02_{Hex}$	(not used)	FF_{Hex}/FF_{Hex}
$03_{Hex}/04_{Hex}$	(not used)	FF_{Hex}/FF_{Hex}
$05_{Hex}/06_{Hex}$	(not used)	FF_{Hex}/FF_{Hex}
$00_{Hex}/07_{Hex}$	Invert TI-Bit	

Host:

36.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Set AE-Bit

37.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

38.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

We recommend that you carefully document which parameters are used for start addresses and number of bytes for writing/reading the desired data records.

The data are sequenced in the exact order specified in the program.

Function Description
Examples for protocol sequence

Example No. 8
Use Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Read code tag using Program No. 1 (code tag type with 32 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 21_{Hex}
02_{Hex}	Program number 01_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 0 (32 byte block size), set AV-Bit

3.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Invert TI-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Set AE-Bit

4.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

... A total of 27 bytes of data are exchanged.
For the remainder of the procedure, see Example 4 on 🗋37.

Dynamic mode is turned off while the Mixed Data Access program is being run.

Function Description
Examples for protocol sequence

Example No. 9
Use Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Write code tag using Program No. 1 (code tag type with 32 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 21_{Hex}
02_{Hex}	Program number 01_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 0 (32 byte block size), set AV-Bit

3.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the first 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TI-Bit

BIS C-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit, invert TO-Bit

4.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Copy the first 6 data bytes

Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

... A total of 27 bytes of data are exchanged.
For the remainder of the procedure, see Example 6 on 🗋39.

Dynamic mode is turned off while the Mixed Data Access program is being run.

Function Description
Examples for protocol sequence

Example No. 10

Put the relevant read/write head into ground state:

Both read/write heads can be independently set to the ground state.

Host:

1.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Set GR-Bit

BIS C-60_2 Identification System:

2.) Go to ground state;
Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset BB-Bit

3.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset GR-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Set BB-Bit

Read/Write Times

Read times from code tag to processor in static mode
(parametering:
2nd byte, bit 5 = 0)

For double read and compare:

Data carrier with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 31	110
for each additional 32 bytes add	120
from 0 to 255	= 950

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 63	220
for each additional 64 bytes add	230
from 0 to 2047	= 7350

Write times from processor to code tag in static mode
(parametering:
2nd byte, bit 5 = 0)

Including readback and compare:

Data carrier with 32 byte blocks	
No. of bytes	Write time [ms]
from 0 to 31	110 + n * 10
for 32 bytes or more	y * 120 + n * 10

Data carrier with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 63	220 + n * 10
for 64 bytes or more	y * 230 + n * 10

n = number of contiguous bytes to write
y = number of blocks to be processed

Example: 17 bytes from address 187 have to be written. Code tag with 32 bytes per block. The blocks 5 and 6 will be processed since the start address 187 is in block 5 and the end address 203 in block 6. t = 2 * 120 + 17 * 10 = 410 ms

The indicated times apply after the code tag has been recognized. If the code tag is not yet recognized, an additional 45 ms for building the required energy field until the code tag is recognized must be added.

Read/Write Times

Read times from code tag to processor in dynamic mode (parametering: 2nd byte, bit 5 = 1)

Read times within the 1st block for dual read and compare:

The indicated times apply after the code tag has been recognized. If the code tag is not yet recognized, an additional 45 ms for building the required energy field until the code tag is recognized must be added.

Data carrier with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 3	14
for each additional byte add	3.5
from 0 to 31	112

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 3	14
for each additional byte add	3.5
from 0 to 63	224

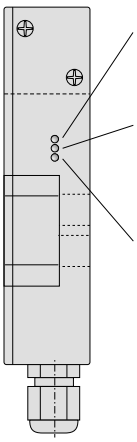
m = highest address to be read

Formula: $t = (m + 1) * 3.5$ ms

Example: Read 11 bytes starting at address 9, i.e. the highest address to be read is 19. This corresponds to 70 ms.

LED Display

Function displays on BIS C-60_2



The BIS C-60_2 uses the three side-mounted LED's to indicate important conditions of the identification system.

Status	LED	Meaning
Ready / Bus active	red	Supply voltage OK; no hardware error, however, bus not active.
	green	Supply voltage / hardware OK, bus active.
CT1 Present / operating	green	Code tag read/write-ready at read/write head 1.
	yellow	Read/write command at read/write head 1 in process.
	yellow flashes	Cable break to read/write head or not connected.
	off	No code tag in read/write range of read/write head 1.
CT2 Present / operating	green	Code tag read/write-ready at read/write head 2.
	yellow	Read/write command at read/write head 2 in process.
	yellow flashes	Cable break to read/write head or not connected.
	off	No code tag in read/write range of read/write head 2.

If all three LED's are synchronously flashing, it means a hardware error. Return the unit to the factory.

BIS C-6002
Mounting Head / Processor

Orientation of the read/write head or adapter

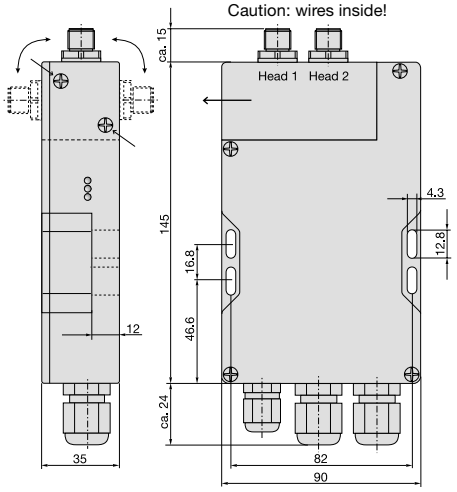
Depending on model, the processor is equipped with a read/write head or the adapter for offset read/write heads. Both the read/write head and the adapter can be rotated by the user by + or –90 deg. to the desired position (see drawing). Be sure that power is off first. Loosen both screws (indicated with arrows). Carefully pull the head or adapter out towards the side (direction of arrow, right drawing).

Caution: wires inside!

Reattach at the desired orientation and screw tight again.

Mounting the BIS C-6002 processor

The processor is attached using 4 M4 screws.



BIS C-6002
Opening the Processor

Opening the Processor BIS C-6002

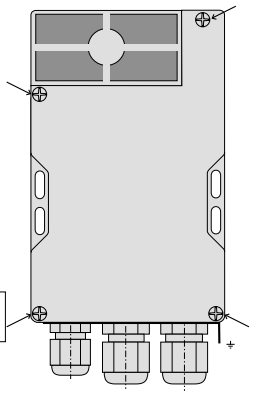
The BIS C-6002 processor must be opened to perform the following steps:

- Set PROFIBUS-DP address
- Activate/deactivate termination resistor
- Set/change compatibility mode
- Replace EEPROM
- Make electrical connections (supply voltage, in-/output, PROFIBUS-DP connections).

Be sure that the unit is disconnected from power before opening.

Remove the 4 screws on the BIS C-6002 and lift off the cover.

Perform the desired action. To make the electrical connections, push the cables through the fittings. For additional wiring details, see the following 🕮.



Opening the processor

BIS C-6002
Installing the connection cables

Make connections on the BIS C-6002 processor

The BIS C-6002 processor must be opened in order to make the connections for the supply voltage, the digital input and the PROFIBUS connections (see 50).

First be sure that the unit is disconnected from power.

Remove the 4 screws on the BIS C-6002 and lift off the cover.

Guide the two PROFIBUS cables through the PG 11 fittings (see 52). For additional information on wiring, see the following .

Push the cable for supply voltage and for the digital input through the PG 9 fitting.

Close up the processor.

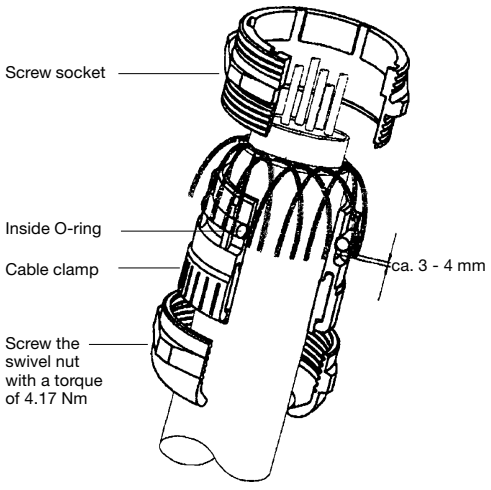
If the processor is equipped with an adapter:

– BIS C-650: Connect the read/write heads to terminals Head 1 and Head 2.
– BIS C-670: Connect the read/write head to terminal Head 1.

BIS C-6002
Mounting the PG Connection for PROFIBUS-DP

Connecting the shield of the PROFIBUS-DP cable in the PG 11 housing on the processor BIS C-6002

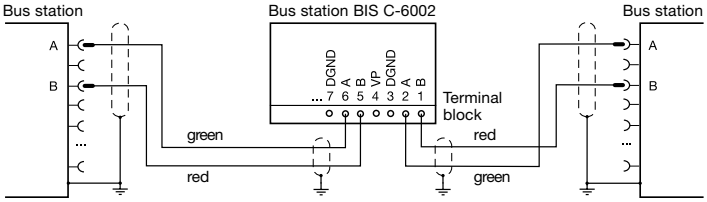
After connecting the (field) bus leads to the termional block, make sure that the shield has proper connection to the PG housing.



BIS C-6002
Interface Information / Wiring Diagrams

Remote bus cable and interfaces for PROFIBUS-DP

To insert BIS C-6002 processor into the serial PROFIBUS, terminals 1and 2 and 5 and 6 for the PROFIBUS-DP interface are located on the terminal block ("incoming" and "outgoing").



In case the processor is the last bus module in the chain, then only the incoming cable is connected. You can either use the connections 1 and 2 or 5 and 6.

The last bus module must terminate the bus with a resistor. In the case of the BIS C-6002, this can be realized in two different ways:

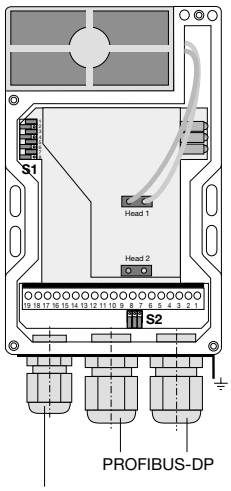
1. **In the device**
   by closing the switch S2
   (factory standard is open)

S2	Terminating resistor
closed	active
open	passive

2. **Outside the device** in a plug. In this case the signals VP (terminal 4) and DGND (terminal 3) should be brought out in order to connect the external resistor to the potential.
   Note: In this case S2 has to be open!

BIS C-6002
Interface Information / Wiring Diagrams

Wiring diagram for BIS C-6002 processor with integrated read/write head



Terminal location and designation

7	6	5	4	3	2	1
DGND	A	B	VP	DGND	A	B
PROFIBUS-DP						

13	12	11	10	9	8
+IN	–IN	$+V_{SO}$	$-V_{SO}$	01	02
INPUT		OUTPUT			

01 = CT Present 1
02 = CT Present 2

Terminal block

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	GND
POWER			RS 232		

Protection ground PE

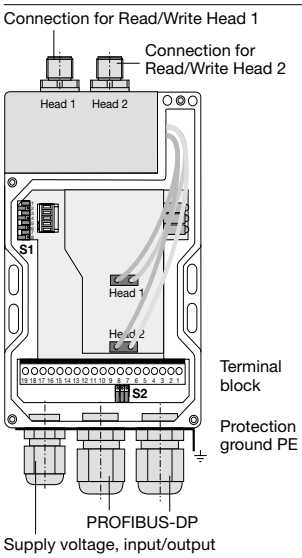
Terminal block connections

The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).

When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on 52.

BIS C-6002
Interface Information / Wiring Diagrams

Wiring diagram for BIS C-6002 processors with BIS C-650 adapter



Terminal location and designation

7	6	5	4	3	2	1
DGND	A	B	VP	DGND	A	B
PROFIBUS-DP						

13	12	11	10	9	8
+IN	–IN	$+V_{so}$	$-V_{so}$	01	02
INPUT		OUTPUT			

01 = CT Present 1
02 = CT Present 2

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	GND
POWER			RS 232		

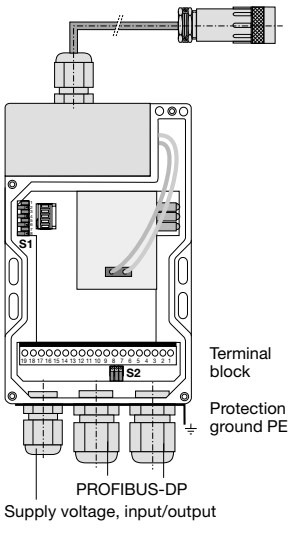
Terminal block connections

The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).

When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on 🗋 52.

BIS C-6002
Interface Information / Wiring Diagrams

Wiring diagram for BIS C-6002 processors with BIS C-670 adapter



Terminal location and designation

7	6	5	4	3	2	1
DGND	A	B	VP	DGND	A	B
PROFIBUS-DP						

13	12	11	10	9	8
+IN	–IN	$+V_{so}$	$-V_{so}$	01	02
INPUT		OUTPUT			

01 = CT Present 1
02 = CT Present 2

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	GND
POWER			RS 232		

Terminal block connections

The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).

When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on 🗋 52.

BIS C-6002
Changing the EEPROM

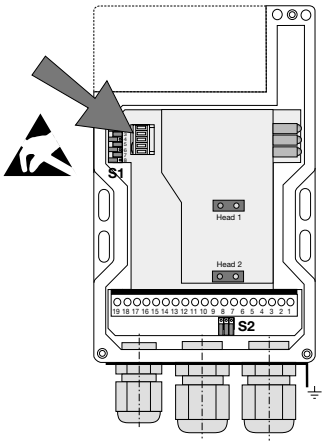
Changing the EEPROM in the BIS C-6002 processor

To replace the EEPROM, open up the processor as described on 50.

Be sure before opening that the unit is disconnected from power..

To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.

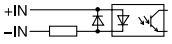
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

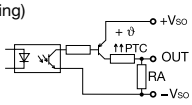
BIS C-6002
Technical Data

Dimensions, Weight	**Housing**	Plastic ABS
	Dimensions with read/write head BIS C-65_	ca. 169 x 90 x 35 mm
	Dimensions with adapter BIS C-650	ca. 185 x 90 x 35 mm
	Weight	ca. 500 g
Operating Conditions	Ambient temperature	0 °C to + 50 °C
Enclosure Rating	Enclosure rating	IP 65 (with read/write head)
Connections	Terminal block	19-pin
	Cable entry	2 x PG 11 fittings (metal)
	Cable diameter	5 to 10 mm
	Cable entry	1 x PG 9 fittings (metal)
	Cable diameter	4 to 8 mm
	Conductor size	0.14 to 1 mm^2
	with ferrules	0.25 to 0.34 mm^2
Electrical Connections	**Supply voltage V_S, input**	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Current draw	≤ 400 mA
	PROFIBUS-DP slave	Terminal block, electrically isolated
	Digital Input (+IN, –IN)	Terminal block, Optocoupler isolated
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to –40 V
	Input current at 24 V	11 mA
	Delay time, typ.	5 ms



BIS C-6002
Technical Data

Electrical Connections (continued)	**Control outputs (01 and 02)** (01 = CT Present 1, 02 = CT Present 2)	Terminal block, Optocoupler isolated
	Output circuit	PNP (current sourcing)
	Operating voltage V_{SO} (external) for output	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Output current	max. 20 mA
	Voltage drop at 20 mA	approx. 2.5 V
	Output resistance R_A	10 kΩ gegen $-V_{SO}$
	Service interface	RS 232
	Read/Write Head	integrated, BIS C-65_ and following *);
	option for mounted adapter BIS C-650 *)	2 x connectors 4-pin (male) for all read/writ heads BIS C-3_ _ with 4-pin connector (female), **except** BIS C-350 and BIS C-352
	option for mounted adapter BIS C-670 *)	1 x connector 8-pin (male) for one of the read/write heads BIS C-350 or BIS C-352
Function Displays	BIS operating messages:	
	Ready / Bus active	LED red / green
	CT1 Present / operating	LED green / yellow
	CT2 Present / operating	LED green / yellow



*) rotatable by 90 degrees

BIS C-6002
Technical Data

The CE-Mark is your assurance that our products are in conformance with the EC-Guideline

89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard

EN 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

BIS C-6002
Ordering Information

Ordering Code

BIS C-6002-028-_ _ _-03-KL2

Balluff Identification System

Type C Read/Write System

Hardware Type
6002 = plastic housing, PROFIBUS-DP

Software-Type
028 = PROFIBUS-DP with memory optimization

Read/Write Head
000 = no read/write head
651 = with read/write head Type 651 (with circular antenna on top)
652 = with read/write head Type 652 (with circular antenna on front)
653 = with read/write head Type 653 (with rod antenna)
650 = adapter with two connections for external read/write heads BIS C-3_ _ (except BIS C-350 and -352)
670 = adapter with one cable connection for an external read/write head BIS C-350 or BIS C-352)

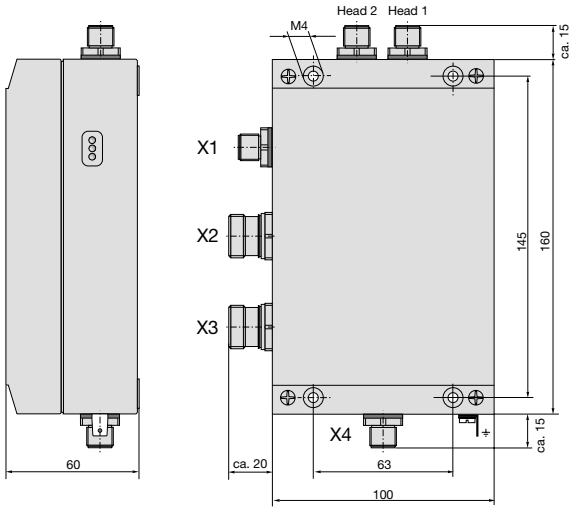
Interface
03 = bus versions

User Connection
KL2 = terminal block via 2 x PG 11 and 1 x PG 9 cable fittings

BIS C-6022
Mounting Processor

Mounting the BIS C-6022 processor

The processor is mounted using 4 M4 screws.



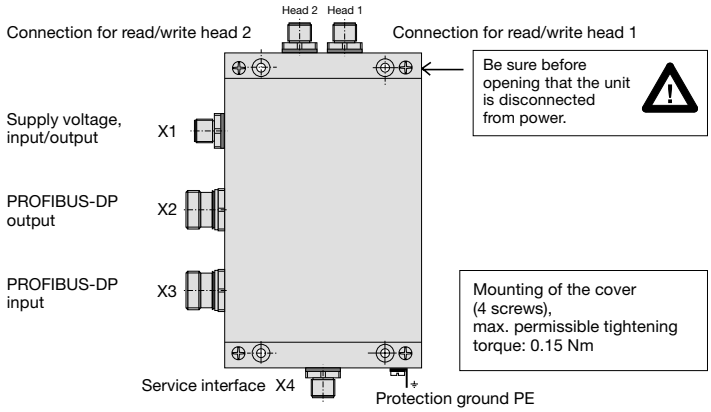
BIS C-6022
Opening the processor / Interface information

Opening the BIS C-6022 processor

To set the PROFIBUS-DP address, activate or deactivate the internal termination resistor, set the compatibility mode or to change the EEPROM, you must open up the BIS C-6022 processor.

Remove the 4 screws on the BIS C-6022 and lift off the cover. See the following 🕮 for additional information.

BIS C-6022 interfaces



Connection locations and names

BIS C-6022
Interface Information / Wiring Diagrams

To insert BIS C-6022 processor into the serial PROFIBUS and to connect the supply voltage and the digital input, the cables have to be connected to the terminals of the processor. For more details regarding the wiring see the following 🕮. The read/write heads have to be connected to the to terminals Head 1 and Head 2.

PROFIBUS-DP

Ensure that the device is turned off.

Connect the "incoming" PROFIBUS cable to the PROFIBUS-Input. Connect the "outgoing" PROFIBUS cable to the PROFIBUS-Output.

The last bus module must terminate the bus with a resistor. In the case of the BIS C-6022, this can be realized in two different ways:

1. **In the device**
   by closing the switch S2
   (factory standard is open)
   The PROFIBUS-Output must
   be closed off with a screw cover in order to maintain the enclosure rating.

S2	Terminating resistor
closed	active
open	passive

2. **Outside the device** in a connector. In this case the signal VP and DGND should be brought out in order to connect the external resistor to the potential.
   Note: In this case the S2 switch has to be open!

No supply voltage is allowed on the PROFIBUS connections!

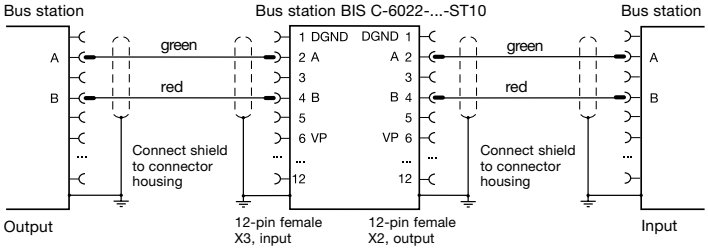
Connect cable for the supply voltage, the digital input, and the outputs to terminal X1.

BIS C-6022
Interface Information / Wiring Diagrams

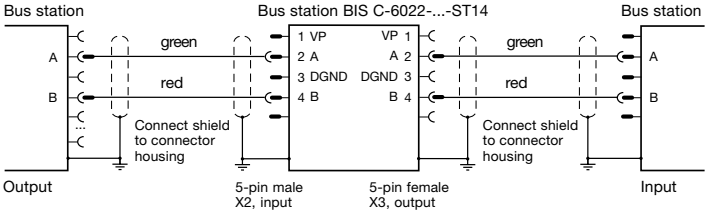
Remote bus cable for PROFIBUS-DP

To insert BIS C-6022 processor into the serial PROFIBUS-DP, there are the terminal X2 for the PROFIBUS output and the terminal X3 for the PROFIBUS input.

BIS C-6022-...-ST10

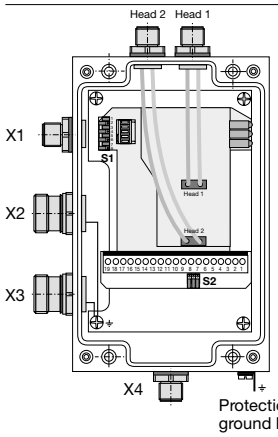


BIS C-6022-...-ST14



BIS C-6022-...-ST10
Interface Information / Wiring Diagrams

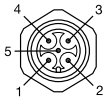
Wiring diagram for BIS C-6022-...-ST10 processor



The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).

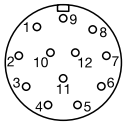
When connecting the bus leads, make sure that the shield has proper connection to connector housing.

X1, supply voltage, digital input, and CT Present outputs



Pin	Function
1	+Vs
2	CT Present 2
3	–Vs
4	+IN
5	CT Present 1

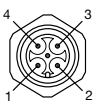
X2, PROFIBUS output
X3, PROFIBUS input



Pin	Function
1	DGND
2	A
3	n.c.
4	B
5	n.c.
6	VP
7	+24 V
8	GND
9	⏚
10 ... 12	n.c.

n.c. = do not connect!

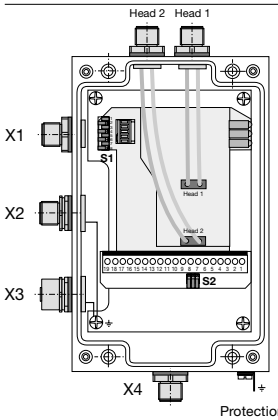
X4, Service interface



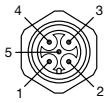
Pin	Function
1	n.c.
2	TxD
3	GND
4	RxD

BIS C-6022-...-ST14
Wiring Diagram

Wiring diagram for BIS C-6022-...-ST14 processor

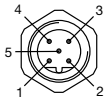


X1, supply voltage, digital input

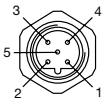


Pin	Funktion
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X2, PROFIBUS input (male)

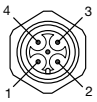


X3, PROFIBUS output (female)



Pin	Funktion
1	VP
2	A
3	DGND
4	B
5	n.c.

X4, Service interface



Pin	Function
1	n.c.
2	TxD
3	GND
4	RxD

n.c. = do not connect!

The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).
When connecting the bus leads, make sure that the shield has proper connection to connector housing.

BIS C-6022
Changing the EEPROM

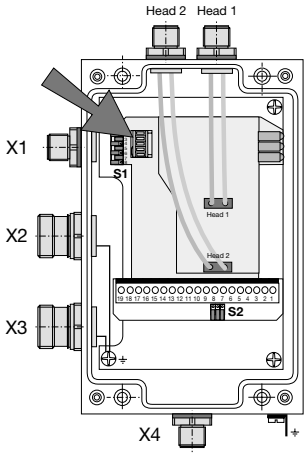
Changing the EEPROM in the BIS C-6022 processor

To change the EEPROM, open the processor as described on □ 63.

Be sure before opening that the unit is disconnected from power.

To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.

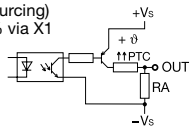
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

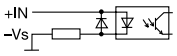
BIS C-6022
Technical Data

Dimensions, weight	**Housing**	Metal
	Dimensions	190 x 120 x 60 mm
	Weight	820 g
Operating conditions	Ambient temperature	0 °C to +60 °C
Enclosure	Protection class	IP 65 (when connected)
Connections BIS C-6022-...-ST10	Integral connector X1 for **V_S, CT Present_, +IN**	5-pin (male)
	Round connector X2 / X3 for **PROFIBUS-DP**	12-pin (female)
	Integral connector X4 for **Service interface**	4-pin (male)
Connections BIS C-6022-...-ST14	Integral connector X1 for **V_S, +IN**	5-pin (male)
	Integral connector X2 for **PROFIBUS-DP** input	5-pin (male)
	Integral connector X3 for **PROFIBUS-DP** output	5-pin (female)
	Integral connector X4 for **Service interface**	4-pin (male)
Electrical connections	**Supply voltage V_S**	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Current draw	≤ 400 mA
with ST10 only:	**Control outputs CT Present 1 and 2**	Optocoupler isolated
	Output circuit	PNP (current sourcing)
	Operating voltage V_S for output	DC 24 V ± 20 % via X1
	Ripple	≤ 10 %
	Output current	max. 20 mA
	Voltage drop at 20 mA	approx. 2.5 V
	Output resistance R_A	10 kΩ to $-V_S$



BIS C-6022
Technical Data

Electrical Connections (continued)	**Digital input +IN**	Optocoupler isolated
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to –40 V
	Input current at 24 V	11 mA
	Delay time, typ.	5 ms
	PROFIBUS-DP, Connector X2, X3	serial interface for PROFIBUS stations
	Head 1, Head 2, Read/Write Head	via 2 x connectors for all read/write heads BIS C-3_ _ with 4-pin connector (female), **excluding** BIS C-350 and BIS C-352
	Service interface X4	RS 232
Function displays	BIS operating messages:	
	Ready / Bus active	LED red / green
	CT1 Present / operating	LED green / yellow
	CT2 Present / operating	LED green / yellow



The CE-Mark is your assurance that our products are in conformance with the EC-Guideline

89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard

EN 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

BIS C-6022
Ordering Information

Ordering code

BIS C-6022-028-050-03-ST_ _

Balluff Identification System

Type C Read/Write System

Hardware Type
6022 = metal housing, PROFIBUS-DP

Software Type
028 = PROFIBUS-DP with memory optimization

Version
050 = with two connections for external read/write heads BIS C-3_ _
(except BIS C-350 and -352)

Interface
03 = bus versions

User Connection
ST10 = Connector version X1, X2, X3, X4 (male: 1× 5-pin, 1× 4-pin, female: 2× 12-pin)
ST14 = Connector version X1, X2, X3, X4 (male: 2× 5-pin, 1× 4-pin, female: 1× 5-pin)

Accessory
(optional,
not included)

Type		Ordering code for ST10	Ordering code for ST14
Mating connector	for X1	BKS-S 79-00	BKS-S 79-00
	for X2	BKS-S 86-00	BKS-S103-00
	for X3	BKS-S 86-00	BKS-S105-00
	for X4	BKS-S 10-3	BKS-S 10-3
Protective cap	for Head_, X4	BES 12-SM-2	BES 12-SM-2
Protective cap		115 475 for X2	BKS 12-CS-01 for X3
Termination			BKS-S105-R01 for X3

Appendix, ASCII Table

Decimal	Hex	Control Code	ASCII
0	00	Ctrl @	NUL
1	01	Ctrl A	SOH
2	02	Ctrl B	STX
3	03	Ctrl C	ETX
4	04	Ctrl D	EOT
5	05	Ctrl E	ENQ
6	06	Ctrl F	ACK
7	07	Ctrl G	BEL
8	08	Ctrl H	BS
9	09	Ctrl I	HT
10	0A	Ctrl J	LF
11	0B	Ctrl K	VT
12	0C	Ctrl L	FF
13	0D	Ctrl M	CR
14	0E	Ctrl N	SO
15	0F	Ctrl O	SI
16	10	Ctrl P	DLE
17	11	Ctrl Q	DC1
18	12	Ctrl R	DC2
19	13	Ctrl S	DC3
20	14	Ctrl T	DC4
21	15	Ctrl U	NAK
22	16	Ctrl V	SYN
23	17	Ctrl W	ETB
24	18	Ctrl X	CAN
25	19	Ctrl Y	EM
26	1A	Ctrl Z	SUB
27	1B	Ctrl [	ESC
28	1C	Ctrl \	FS
29	1D	Ctrl]	GS
30	1E	Ctrl ^	RS
31	1F	Ctrl _	US
32	20		SP
33	21		!
34	22		"
35	23		#
36	24		$
37	25		%
38	26		&
39	27		'
40	28		(
41	29		)
42	2A		*
43	2B		+
44	2C		,
45	2D		-
46	2E		.
47	2F		/
48	30		0
49	31		1
50	32		2
51	33		3
52	34		4
53	35		5
54	36		6
55	37		7
56	38		8
57	39		9
58	3A		:
59	3B		;
60	3C		<
61	3D		=
62	3E		>
63	3F		?
64	40		@
65	41		A
66	42		B
67	43		C
68	44		D
69	45		E
70	46		F
71	47		G
72	48		H
73	49		I
74	4A		J
75	4B		K
76	4C		L
77	4D		M
78	4E		N
79	4F		O
80	50		P
81	51		Q
82	52		R
83	53		S
84	54		T
85	55		U
86	56		V
87	57		W
88	58		X
89	59		Y
90	5A		Z
91	5B		[
92	5C		\
93	5D		]
94	5E		^
95	5F		_
96	60		`
97	61		a
98	62		b
99	63		c
100	64		d
101	65		e
102	66		f
103	67		g
104	68		h
105	69		i
106	6A		j
107	6B		k
108	6C		l
109	6D		m
110	6E		n
111	6F		o
112	70		p
113	71		q
114	72		r
115	73		s
116	74		t
117	75		u
118	76		v
119	77		w
120	78		x
121	79		y
122	7A		z
123	7B		{
124	7C		|
125	7D		}
126	7E		~
127	7F		DEL